

Fondamenti di Informatica L-A (A.A. 2002/2003) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d’Esame di Martedì 08/07/2003 – durata 2h:30m

ESERCIZIO 1 (12 punti)

Si scriva una procedura **p2()** che si occupi di popolare un file di testo chiamato **sport.txt** (2 punti).

P2() ha come parametro di ingresso un puntatore a file e deve richiedere all’utente l’inserimento interattivo da tastiera di una serie di dati da salvare sul file il cui puntatore è passato come parametro. Il file di testo dovrà contenere l’elenco dei prodotti di abbigliamento sportivo disponibili in un negozio. Si preveda di terminare l’inserimento dati digitando, ad esempio, ‘0’ come sesso del cliente. Più precisamente, ogni riga del file include, nell’ordine:

- sesso del cliente target (carattere uguale a ‘f’ per abbigliamento femminile, o a ‘m’ per abbigliamento maschile), uno e un solo spazio di separazione;
- codice identificativo del capo di abbigliamento (numero intero), uno e un solo spazio di separazione;
- prezzo in euro del capo di abbigliamento (numero float), uno e un solo spazio di separazione;
- descrizione del capo (non più di 20 caratteri senza spazi).

Ad esempio, **sport.txt**:

```
m 1772 25.50 shirt
f 7453 47.00 bikini
f 2437 51.90 sandali
m 2332 7.95 guantoni
f 9825 23.00 shirt
.....
```

Si scriva una procedura **p1()** che riceva come parametro di ingresso un puntatore a file di testo e restituisca come parametri di uscita un vettore **y** contenente strutture **capo** (sesso, prezzo, descrizione) e il numero degli elementi **N** inseriti in **y**. **P1()** deve caricare in **y** le informazioni sui soli capi di abbigliamento il cui prezzo sia superiore al prezzo di quello immediatamente precedente nel file (4 punti).

Si scriva un programma C che, utilizzando le procedure **p2()** e **p1()** precedentemente definite, inserisca in un vettore **z** (supposto di dimensione massima **DIM=100**) le informazioni sui capi di prezzo superiore a quelli immediatamente precedente contenuti nel file **sport.txt**. Il programma deve inoltre stampare a terminale tutti gli elementi di **z** relativi ai capi di abbigliamento femminile, purché ce ne siano **almeno tre** in **z** (6 punti).

ESERCIZIO 2 (5 punti)

Si scriva una funzione **f1()** che data in ingresso una lista di interi ed un numero intero **x**, restituisca in uscita una lista, ottenuta dalla lista di partenza aggiungendo eventualmente, **in posizione qualunque**, un nuovo elemento di valore **x**, cosicché le occorrenze di **x** nella lista siano sempre dispari. Ad esempio, se invocata con **l=[11,7,8,0,15,7,-2]** e **x=7**, la funzione **f1()** deve restituire una lista con una occorrenza in più di **x**, ad esempio **[11,7,8,0,15,7,-2,7]**.

La funzione **f1()** può essere realizzata in modo ricorsivo o iterativo, facendo riferimento al tipo di dato astratto **list** e alle sue operazioni primitive definite durante il corso (che quindi possono NON essere riportate nella soluzione), oppure utilizzando direttamente la rappresentazione collegata a puntatori.

ESERCIZIO 3 (7 punti)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione (si motivi opportunamente la risposta data)?

```
#include <stdio.h>
#include <stdlib.h>
#define L 20
void Proc(char [], int);
int N=L;
```

```

main () {
char *s= "Appello di Luglio F";
int i;

Proc(s,L/2);
N+3; N++;
{   int i=0;
    for (i=L/2; i<L; i=i+2) printf("%c\n", *(s-L/2+i));
}
printf("N vale adesso: %d\n",N);
}

void Proc(char y[], int DIM) {
int i, N;
N++; i=1;
while (i<DIM) { y[i]='x'; i=i+3; }
}

```

Esercizio 4 (4 punti)

Si consideri la seguente funzione F:

```

float F(int x){
    if (x!=0) {
        if (x<0) x=-(x+1); else x=-(x-1);
        return F(x/3)*2; }
    else return -1;
}

```

Si scriva il risultato della funzione quando invocata come **F(-7)** e si disegnino i corrispondenti record di attivazione.

Esercizio 5 (2 punti)

Si consideri la grammatica G con scopo S e simboli terminali {**printf,scanf,(,),x,","%,d,,}**

```

S ::= A E D | E B
D ::= C C B | C B B
E ::= D D | B B D
A ::= printf | scanf
B ::= ( | ) | x | "
C ::= % | d | ,

```

Si dica se la stringa **printf ("%d", x)** è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione *left most*.

Esercizio 6 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato, traslandolo poi in decimale per la verifica:

$$39 + 73$$

SOLUZIONE

ESERCIZIO 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DIM 100

typedef struct {char sex; float prezzo; char desc[21];} capo;

void p2(FILE * F) {
    char sex, desc[21]; int codice; float p;
    do {
        printf("Inserire dati da salvare sul file. Per ogni capo: sesso, codice,
        prezzo, descrizione.\n Inserire 0 per terminare");
        scanf("%c %d %f %s\n", &sex, &codice, &p, desc);
        if (sex!='0')
            fprintf(F, "%c %d %s %d\n", sex, codice, p, desc);
    }
    while (sex!='0');
}

void p1(FILE * F, capo y[], int *num) {
    char sex, desc[21]; int codice; float p, prec=0;
    *num=0;
    while (fscanf(F,"%c %d %f %s\n", &sex, &codice, &p, desc) != EOF)
        { if (p>prec)
            { prec=p;
              y[*num].sex=sex;
              y[*num].prezzo=p;
              strcpy(desc, y[*num].desc);
              *num = *num+1; } }
}

main() {
    capo z[DIM]; int N,i,j,stampabile=0; FILE* f;

    if ((f=fopen("c:\\sport.txt", "w"))==NULL) {
        printf("Non sono riuscito ad aprire il file in scrittura!"); exit(1); }
    p2(f);
    fclose(f); f=fopen("c:\\sport.txt", "r");
    p1(f, z, &N);
    for (i=0; i<N; i++)
        if (z[i].sex=='f') stampabile++; //ottimizzazione del ciclo?
    if (stampabile>2)
        { for (i=0; i<N; i++)
            if (z[i].sex=='f')
                printf("%c %f %s\n", z[i].sex, z[i].prezzo, z[i].desc);
        }
    fclose(f);
}
```

ESERCIZIO 2

```
list f1(list l, int x) {
    int count=0; list temp=l;

    while (temp!=NULL) {
```

```

    if (temp->value==x) count++; temp=temp->next; }
if (count%2==0) return cons(x,1);
    else return 1;
}

```

ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:

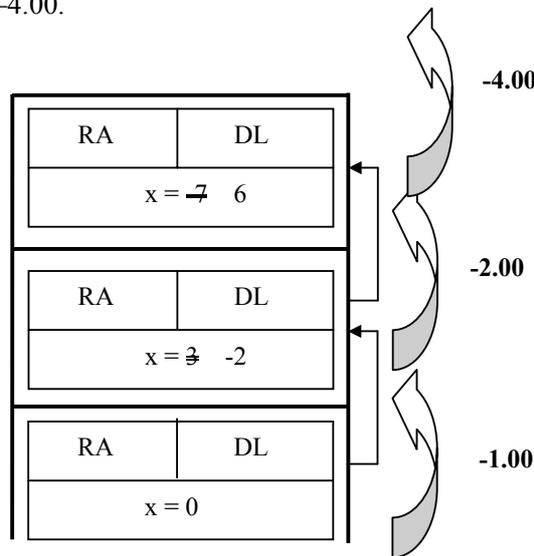
A p x o d (ogni carattere su riga differente)

N vale adesso: 21

Infatti la stringa s viene inizializzata a “Appello di Luglio F”, terminatore compreso. Successivamente viene invocata la procedura Proc(), di cui è stato dichiarato il prototipo prima del main(), alla quale si passa per riferimento la stringa (vettore di char). In particolare, Proc mette a ‘x’ i caratteri della stringa di posizione 1, 4, e 7. Infine, il programma principale stampa a video alcuni caratteri della stringa s, in particolare quelli di posizione pari, a partire da 0 fino a 8, estremi inclusi.

ESERCIZIO 4

La funzione restituisce il valore -4.00.



ESERCIZIO 5

La stringa è sintatticamente corretta. Derivazione *left most*:

```

s → A E D → printf E D → printf B B D D → printf (" C C B D →
→ printf ("%d" C B B → printf ("%d", x)

```

ESERCIZIO 6

```

39-> 0 0100111
73-> 0 1001001

```

Tra i moduli dei numeri si esegue una addizione ottenendo: 0 1110000
che vale 112 in base dieci.